

2

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 80-0187	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FAULT ISOLATION OF MODULAR EQUIPMENT WITH IMPERFECT BUILT-IN-TESTS		5. TYPE OF REPORT & PERIOD COVERED FINAL
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Theodore J. Sheskin		8. CONTRACT OR GRANT NUMBER(s) AFOSR-78-3496
9. PERFORMING ORGANIZATION NAME AND ADDRESS Cleveland State University Department of Industrial Engineering Cleveland, Ohio 44115		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A6
11. CONTROLLING OFFICE NAME AND ADDRESS AFOSR/NM Bolling AFB, Washington, DC 20332		12. REPORT DATE 1979
		13. NUMBER OF PAGES 47
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Was restricted but changed to non-restrictive and sent to DTIC. Approved by Dr. Bram on 1-11-84.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <div style="text-align: right;">DTIC ELECTE JAN 20 1984 S D E</div>		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The objective of this research is to derive a minimum cost sequence of automatic built-in-tests (BITs) which will partition modular equipment into mutually exclusive groups of modules. Following an equipment malfunction, one of these groups will be identified by the BIT diagnostic subsystem as the group which contains the faulty module. The BITs will not detect all of the possible errors in the modules, and they may also generate false alarms by calling out a group of		

unclassified

AD A137046

WIL FILE COPY

Cont'd. #20 Abstract:

modules which does not contain the faulty unit. Both the cost of a BIT and the probability that a BIT will pass or fail are functions of the modules which are tested. A recursive algorithm is developed which consists of a backward computational process followed by a forward computational process. The recursive algorithm generates a sequence of BITs with a minimum cost. The algorithm is applied to a four-module sample problem to produce a numerical solution.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



unclassified

AFOSR-TR- 80 - 0187

Information
15 Jan. 1979

Final Report

Air Force Office of Scientific Research

Air Force Systems Command, USAF

Grant No. AFOSR-78-3496

Fault Isolation of Modular Equipment
with Imperfect Built-in-Tests

Theodore J. Sheskin

Dept. of Industrial Engineering

Cleveland State University

Cleveland, Ohio 44115

216 / 687-2044

01 19 072

Approved for public release;
distribution unlimited.

-1-

Abstract

The objective of this research is to derive a minimum cost sequence of automatic built-in-tests (BITs) which will partition modular equipment into mutually exclusive groups of modules. Following an equipment malfunction, one of these groups will be identified by the BIT diagnostic subsystem as the group which contains the faulty module. The BITs will not detect all of the possible errors in the modules, and they may also generate false alarms by calling out a group of modules which does not contain the faulty unit. Both the cost of a BIT and the probability that a BIT will pass or fail are functions of the modules which are tested. A recursive algorithm is developed which consists of a backward computational process followed by a forward computational process. The recursive algorithm generates a sequence of BITs with a minimum ~~life cycle~~ cost. The algorithm is applied to a four-module sample problem to produce a numerical solution.

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
NOTICE OF TRANSMITTAL TO DTIC
This technical report has been
approved for public release
Distribution unlimited.
MATTHEW J. KERRICK
Chief, Technical Information Division

INTRODUCTION

This is the second and final report summarizing research performed under a one year minigrant, Grant No. AFOSR-78-3496, awarded by the Air Force Office of Scientific Research, Air Force Systems Command, USAF. It is a continuation of the research described in the first report, "Specification of Built-in-Tests for Modular Equipment," (19), mailed on May 2, 1978, to Mr. Jerome Klion, RADC/RBRT, Griffis AFB, NY 13441, and to Dr. Joseph Bram, AFOSR/NM, Bolling AFB, DC 20332. The first report was also sent on July 25, 1978, to Mr. I. L. Krulac, RADC/RBRAC, Griffiss AFB, NY 13441. General background information, an introductory discussion of the basic problem, definitions, and diagrams are contained in that earlier report.

In modular airborne and ground electronic equipment, "built-in-test (BIT) diagnostic subsystems" are being used more because they allow fewer and less qualified maintenance personnel, and fewer pieces of external test equipment, which are generally quite expensive.

A primary equipment is composed of modular line replaceable units (LRUs), all of which can be assumed to operate independently. The LRUs have sufficiently low probabilities of failure so that the probabilities of multiple failures can be neglected. Whenever the equipment malfunctions, a single LRU is assumed to have failed, and the BIT will automatically execute a fixed sequence of diagnostic tests to identify the group of LRUs which contains the faulty unit. Secondary isolation will be performed by semi-automatic or manual means, which incur time and other equipment costs, to isolate the failed LRU. The defective unit will be removed and replaced, and the system retested.

The basic objective is to specify a sequence of automatic BITs that will isolate a faulty LRU at a minimum life cycle cost (LCC). Unlike the first report,

in which the BITs could detect all of the equipment errors, the BITs in this report are assumed to be imperfect in the sense that they will not detect all possible errors in the LRUs which they test. Furthermore, the BITs can also give false alarms by erroneously indicating faults in LRUs which are functioning correctly.

A recursive algorithm is developed to generate a minimum cost sequence of automatic BITs. The BIT testing sequence partitions the equipment into mutually exclusive groups of LRUs. Following an equipment malfunction, one of these mutually exclusive groups of LRUs will be identified by the automatic BIT diagnostic as the group which contains the faulty LRU. The recursive algorithm is applied to a four-element sample problem to derive a minimum cost BIT testing sequence.

The Air Force can impose a number of constraints on a BIT testing sequence. In this report the following three representative constraints are described.

1. The BIT subsystem must be capable of detecting a minimum proportion of equipment faults.
2. A specified percentage of the repair times required to perform secondary isolation must not exceed a given value.
3. The mean time to repair the equipment must not exceed a specified value.

NOTATION

N	number of LRUs in the equipment
K	number of mutually exclusive groups of LRUs into which the equipment is partitioned by its BIT diagnostic subsystem
BIT	built-in-test
LRU	line replaceable unit
P_i	prior probability of failure of LRU_i , given an equipment malfunction
P_j	probability that the BIT diagnostic identifies group j as the group which contains the single failed LRU
MG_j	number of maintenance manhours required to perform secondary isolation of group j , given that the BIT diagnostic has identified group j as the group which contains the faulty LRU
ME	expected number of maintenance manhours required to perform secondary isolation
MF	expected number of maintenance manhours required to troubleshoot the entire equipment in the event that the BIT diagnostic does not recognize that a failure has occurred
M	mean time to repair the equipment
$(1-P_F)$	proportion of equipment faults detectable by the BIT diagnostic

LCC life cycle cost.

M_{\max} value of maintenance manhours below which a specified percentage, $m\%$, of all secondary isolation maintenance actions should be completed; m^{th} percentile of the distribution of the random variable MG_j .

n number of untested LRUs.

T_k built-in-test k . A test is represented by an N -bit number containing only the bits 0 and 1. A 0 is assigned in position i of a test if LRU_i must be good in order for the test to pass. A 1 is placed in position i of a test if LRU_i is not tested.

C_k cost associated with built-in-test T_k .

S state of the equipment prior to performing the test T_k . A state is represented by an N -bit number containing only the bits 0 and 1. The N bits in the designation of a state correspond, sequentially from left to right, to LRU_1 , LRU_2 , . . . , LRU_N . A 0 is assigned in position i of a state if LRU_i is known to be good. A 1 is assigned in position i of a state if LRU_i is not yet tested. In the initial state there are 1's in all positions since none of the LRUs have been tested.

Y state of the equipment if built-in-test T_k passes. This state is computed by multiplying S and T_k bit by bit with no carry.

Z state of the equipment if built-in-test T_k fails. This state is computed by multiplying S and \bar{T}_k , the complement of T_k , bit by bit without carry.

- \bar{T} complement of built-in-test T .
- S_t a terminal state, which requires secondary isolation to find the single failed unit.
- $I_j(S_t)$ cost of secondary isolation of terminal state S_t , which corresponds to group j of LRUs.
- Y_t a terminal state produced if built-in-test T_k passes.
- Z_t a terminal state produced if built-in-test T_k fails.
- $f(S, T_k)$ life cycle cost of a sequence of built-in-tests, given that the current state of the equipment is S and built-in-test T_k is performed.
- $f(S)$ minimum life cycle cost, given that the current state of the equipment is S .
- $P(S)$ probability that the BIT testing sequence has identified state S as the state which contains the failed LRU.
- $C_{T_k|S}$ life cycle cost of built-in-test T_k , given that the current state of the equipment is S .
- y $\begin{cases} 1, & \text{if } Y \text{ is a non-terminal state} \\ 0, & \text{if } Y_t \text{ is a terminal state} \end{cases}$
- z $\begin{cases} 1, & \text{if } Z \text{ is a non-terminal state} \\ 0, & \text{if } Z_t \text{ is a terminal state} \end{cases}$

SAMPLE PROBLEM

A four-element sample problem with seven BITs is defined in Tables 1 and 2. For each BIT the following information is provided in Table 1:

- 1) the binary designation of the BIT, 2) the LRU under test, 3) the percentage of faults in the LRU under test detected by the BIT, 4) the probability of a fault in the LRU under test, given that the equipment has

malfunctioned, and 5) the cost of implementing the BIT for the LRU under test.

Table 1
BITS for Individual LRUs in Sample Problem

Binary Designation of Bit	LRU under test	Percentage of faults detected by BIT	Conditional Probability of fault in LRU under test	Cost of BIT for LRU under test
0001	1	70	0.2	\$ 11
	2	85	0.3	6
	3	65	0.4	10
0010	1	90	0.2	5
	2	80	0.3	13
	4	75	0.1	8
0011	1	82	0.2	11
	2	93	0.3	4
0100	1	76	0.2	12
	3	67	0.4	15
	4	84	0.1	7
0101	1	87	0.2	14
	3	72	0.4	9
0110	1	68	0.2	12
	4	81	0.1	8
0111	1	74	0.2	10

The times (in hours) and the costs (in dollars) required for secondary isolation of all potential terminal states, each identified by an index j , are given in Table 2. The cost of secondary isolation is equal to the time required for secondary isolation multiplied by a labor rate of \$10 per hour.

Table 2
Costs of Secondary Isolation for Sample Problem

Group j	LRUs	Potential terminal state, S_t	MG_j (hour)	$I_j(S_t)$
1	1	1000	0.3	\$ 3
2	2	0100	0.7	7
3	3	0010	0.5	5
4	4	0001	0.4	4
5	1,2	1100	2	20
6	1,3	1010	2	20
7	1,4	1001	2	20
8	2,3	0110	2	20
9	2,4	0101	2	20
10	3,4	0011	2	20
11	1,2,3	1110	3	30
12	1,2,4	1101	3	30
13	1,3,4	1011	3	30
14	2,3,4	0111	3	30
15	1,2,3,4	1111	4	40

COMPUTATION OF STATE-DEPENDENT PROBABILITIES
THAT A SINGLE BIT WILL PASS OR FAIL

Both the cost of a BIT and the probability that a BIT will pass or fail are state-dependent because they are both functions of which LRUs are tested. For example, the BIT, T_{0001} , will test LRUs 1, 2, and 3 in state 1111. However, in state 0110, since LRU1 is known to be good, T_{0001} will test only LRUs 2 and 3. In state 0010 only LRU3 will be tested by T_{0001} . The state-dependent cost of a BIT is simply the sum of the costs of implementing the BIT for each of the LRUs under test in the given state. Referring to the costs in Table 1, the cost of T_{0001} in state 0110 is $\$6 + \$10 = \$16$ because LRUs 2 and 3 are tested. For the sample problem specified in Tables 1 and 2, the costs of each BIT with different combinations of LRUs under test are summarized in Table 3.

The cost of a BIT is not the only function which is state-dependent. The probability that a single BIT will pass or fail is also dependent on which LRUs are tested. These probabilities are listed in Table 3 for the BITs in the sample problem. The computations of these state-dependent probabilities in Table 3 are illustrated by the following representative calculations for the single BIT, T_{0001} , which can test any combination of LRUs 1, 2, and 3.

1. First assume that LRUs 1 and 2 are known to be good. Only LRU1 is to be tested.

$$\begin{aligned} P(T_{0001} \text{ fail} | \text{LRU1 tested}) &= P(\text{LRU1 bad})P(T_{0001} \text{ detect fault} | \\ &\text{LRU1 bad}) \\ &= 0.2(0.7) = 0.14 \end{aligned} \tag{1}$$

$$\begin{aligned}
P(T_{0001} \text{ pass} | \text{LRU1 tested}) &= P(\text{LRU1 good}) + \\
&P(\text{LRU1 bad})P(T_{0001} \text{ not detect fault} | \text{LRU1 bad}) \\
&= 0.8 + 0.2(0.3) = 0.86 = 1 - P(T_{0001} \text{ fail} | \text{LRU1 tested})
\end{aligned} \tag{2}$$

2. Next assume that only LRU3 is known to be good. LRUs 1 and 2 are to be tested.

$$\begin{aligned}
P(T_{0001} \text{ fail} | \text{LRU1,2 tested}) &= P(\text{LRU1 bad})P(T_{0001} \text{ detect fault} | \\
&\text{LRU1 bad}) + P(\text{LRU2 bad})P(T_{0001} \text{ detect fault} | \text{LRU2 bad}) \\
&- P(\text{LRU1 bad})P(T_{0001} \text{ detect fault} | \text{LRU1 bad})P(\text{LRU2 bad}) \\
&P(T_{0001} \text{ detect fault} | \text{LRU2 bad}) \\
&= 0.2(0.7) + 0.3(0.85) - 0.2(0.7)0.3(0.85) = .359
\end{aligned} \tag{3}$$

$$\begin{aligned}
P(T_{0001} \text{ pass} | \text{LRU1,2 tested}) &= P(\text{LRU1 good})P(\text{LRU2 good}) \\
&+ P(\text{LRU1 bad})P(T_{0001} \text{ not detect fault} | \text{LRU1 bad})P(\text{LRU2 good}) \\
&+ P(\text{LRU1 good})P(\text{LRU2 bad})P(T_{0001} \text{ not detect fault} | \text{LRU2 bad}) \\
&+ P(\text{LRU1 bad})P(T_{0001} \text{ not detect fault} | \text{LRU1 bad})P(\text{LRU2 bad}) \\
&P(T_{0001} \text{ not detect fault} | \text{LRU2 bad}) \\
&= 0.8(0.7) + 0.2(0.3)0.7 + 0.8(0.3)0.15 + \\
&\quad 0.2(0.3)0.3(0.15) = 0.641 \\
&= 1 - P(T_{0001} \text{ fail} | \text{LRU1,2 tested})
\end{aligned} \tag{4}$$

3. Finally assume that LRUs 1, 2, and 3 are to be tested.

$$\begin{aligned}
P(T_{0001} \text{ fail} | \text{LRU1,2,3 tested}) &= P(\text{LRU1 bad})P(T_{0001} \text{ detect fault} | \\
&\text{LRU1 bad}) \\
&+ P(\text{LRU2 bad})P(T_{0001} \text{ detect fault} | \text{LRU2 bad}) \\
&+ P(\text{LRU3 bad})P(T_{0001} \text{ detect fault} | \text{LRU3 bad}) \\
&- P(\text{LRU1 bad})P(T_{0001} \text{ detect fault} | \text{LRU1 bad})P(\text{LRU2 bad}) \\
&P(T_{0001} \text{ detect fault} | \text{LRU2 bad})
\end{aligned}$$

$$-P(\text{LRU1 bad})P(T_{0001} \text{ detect fault} | \text{LRU1 bad})P(\text{LRU3 bad})$$

$$P(T_{0001} \text{ detect fault} | \text{LRU3 bad})$$

$$-P(\text{LRU2 bad})P(T_{0001} \text{ detect fault} | \text{LRU2 bad})P(\text{LRU3 bad})$$

$$P(T_{0001} \text{ detect fault} | \text{LRU3 bad})$$

$$+P(\text{LRU1 bad})P(T_{0001} \text{ detect fault} | \text{LRU1 bad})P(\text{LRU2 bad})$$

$$P(T_{0001} \text{ detect fault} | \text{LRU2 bad})P(\text{LRU3 bad})$$

$$P(T_{0001} \text{ detect fault} | \text{LRU3 bad})$$

(5)

$$= 0.2(0.7) + 0.3(0.85) + 0.4(0.65) - 0.2(0.7)0.3(0.85)$$

$$- 0.2(0.7)0.4(0.65) - 0.3(0.85)0.4(0.65)$$

$$+ 0.2(0.7)0.3(0.85)0.4(0.65) = 0.526$$

$$P(T_{0001} \text{ pass} | \text{LRU1,2,3 tested}) = P(\text{LRU1 good})P(\text{LRU2 good})P(\text{LRU3 good})$$

$$+ P(\text{LRU1 bad})P(T_{0001} \text{ not detect fault} | \text{LRU1 bad})P(\text{LRU2 good})P(\text{LRU3 good})$$

$$+ P(\text{LRU1 bad})P(T_{0001} \text{ not detect fault} | \text{LRU1 bad})P(\text{LRU2 bad})$$

$$P(T_{0001} \text{ not detect fault} | \text{LRU2 bad})P(\text{LRU3 good})$$

$$+ P(\text{LRU1 bad})P(T_{0001} \text{ not detect fault} | \text{LRU1 bad})P(\text{LRU2 good})P(\text{LRU3 bad})$$

$$P(T_{0001} \text{ not detect fault} | \text{LRU3 bad})$$

$$+ P(\text{LRU1 good})P(\text{LRU2 bad})P(T_{0001} \text{ not detect fault} | \text{LRU2 bad})P(\text{LRU3 good})$$

$$+ P(\text{LRU1 good})P(\text{LRU2 bad})P(T_{0001} \text{ not detect fault} | \text{LRU2 bad})P(\text{LRU3 bad})$$

$$P(T_{0001} \text{ not detect fault} | \text{LRU3 bad})$$

$$+ P(\text{LRU1 bad})P(T_{0001} \text{ not detect fault} | \text{LRU1 bad})P(\text{LRU2 bad})$$

$$P(T_{0001} \text{ not detect fault} | \text{LRU2 bad})P(\text{LRU3 bad})$$

$$P(T_{0001} \text{ not detect fault} | \text{LRU3 bad})$$

$$+ P(\text{LRU1 good})P(\text{LRU2 good})P(\text{LRU3 bad})P(T_{0001} \text{ not detect fault} |$$

$$\text{LRU3 bad})$$

(6)

$$= 0.8(0.7)0.6 + 0.2(0.3)0.7(0.6) + 0.2(0.3)0.3(0.15)0.6$$

$$+ 0.2(0.3)0.7(0.4)0.35 + 0.8(0.3)0.15(0.6)$$

$$+ 0.8(0.3)0.15(0.4)0.35 + 0.2(0.3)0.3(0.15)0.4(0.35)$$

$$+ 0.8(0.7)0.4(0.35) = 0.474$$

$$= 1 - P(T_{0001} \text{ fail} | \text{LRU1,2,3 tested})$$

Using formulas analogous to those developed for T_{0001} , the values in Table 3 are computed for the probabilities that all BITS will pass or fail when one, two, or three LRUs are tested. When specified LRUs are tested by a BIT, the probability that the BIT will pass and the probability that the BIT will fail sum to one. This result, which can be proven algebraically, is apparent in each row of Table 3.

COMPUTATION OF STATE PROBABILITIES IN A BIT TESTING DIAGRAM

In a BIT testing diagram a node represents a state of the equipment. To compute the life cycle cost of a BIT testing diagram, the probabilities of terminal states must be calculated. The probabilities of terminal states are the product of the branch probabilities which produce these states. Branch probabilities are conditioned on the current state which is tested. Upward sloping branches contain the probability that the test on the current state passes; downward sloping branches give the probability that the test on the current state fails. The probability of a designated state is the product of the probabilities on all of the upward sloping and downward sloping branches along the path leading from the initial state to the designated state. More concisely, the probability of a designated state is equal to the probability of the preceding state multiplied by the probability of the outcome of the test which produces the designated state.

Table 3

State-dependent Probabilities and Costs
for BITs in Sample Problem

BIT \bar{T}_k	LRUs under test	$P(T_k \text{ pass} \text{LRUs under test})$	$P(T_k \text{ fail} \text{LRUs under test})$	Cost of BIT for LRUs under test
0001	1	0.86	0.14	\$ 11
	2	0.745	0.255	6
	3	0.74	0.26	10
	1,2	0.641	0.359	17
	1,3	0.636	0.364	21
	2,3	0.551	0.449	16
	1,2,3	0.474	0.526	27
0010	1	0.82	0.18	5
	2	0.76	0.24	13
	4	0.925	0.075	8
	1,2	0.623	0.377	18
	1,4	0.759	0.241	13
	2,4	0.703	0.297	21
	1,2,4	0.576	0.424	26
0011	1	0.836	0.164	11
	2	0.721	0.279	4
	1,2	0.603	0.397	15
0100	1	0.848	0.152	12
	3	0.732	0.268	15
	4	0.916	0.084	7
	1,3	0.621	0.379	27
	1,4	0.777	0.223	19
	3,4	0.671	0.329	22
	1,3,4	0.569	0.431	34
0101	1	0.826	0.174	14
	3	0.712	0.288	9
	1,3	0.588	0.412	23
0110	1	0.864	0.136	12
	4	0.919	0.081	8
	1,4	0.794	0.206	20
0111	1	0.852	0.148	10

In symbolic terms, if S denotes the current state and T_k is the test performed, Y is the next state if T_k passes, and Z is the next state if T_k fails, the following pair of equations will give the probability of the next state:

$$P(Y) = P(S)P(T_k \text{ passes} | S) \quad (7a)$$

$$P(Z) = P(S)P(T_k \text{ fails} | S) \quad (7b)$$

To illustrate the computation of state probabilities, consider the BIT testing sequence $\{T_{0001}, T_{0101}, T_{0010}\}$. The corresponding testing diagram appears in Figure 1.

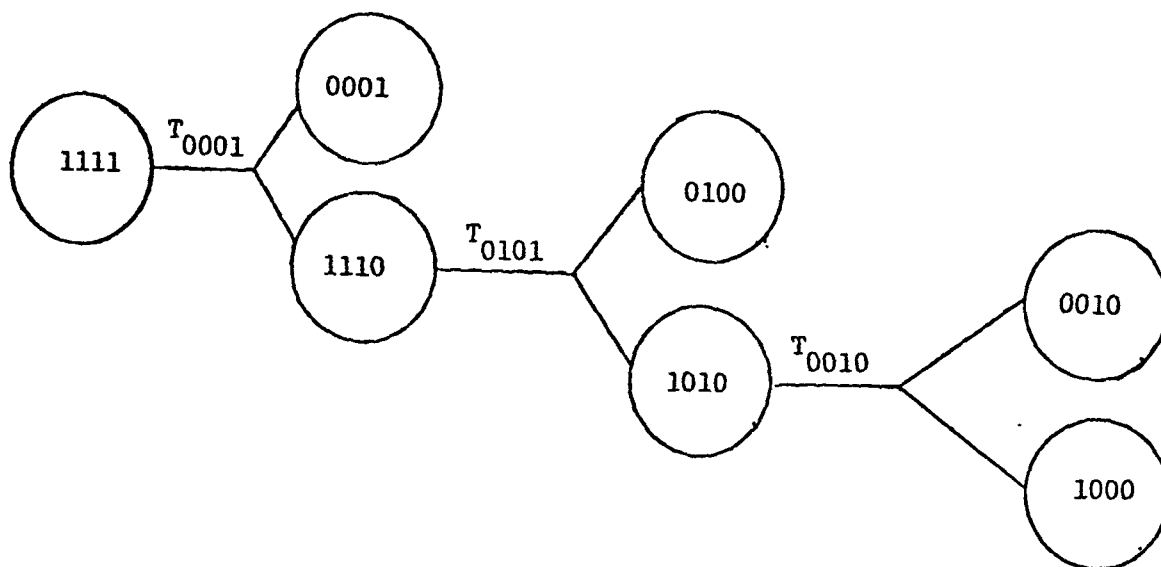


Figure 1. BIT Testing Diagram

The conditional branch probabilities that the BITs will pass or fail have already been computed in Table 3.

$$P(T_{0001} \text{ fail} | 1111) = P(T_{0001} \text{ fail} | \text{LRU1,2,3 tested}) = 0.526$$

$$P(T_{0001} \text{ pass} | 1111) = P(T_{0001} \text{ pass} | \text{LRU1,2,3 tested}) = 0.474$$

$$P(T_{0101} \text{ fail} | 1110) = P(T_{0101} \text{ fail} | \text{LRU1,3 tested}) = 0.412$$

$$P(T_{0101} \text{ pass} | 1110) = P(T_{0101} \text{ pass} | \text{LRU1,3 tested}) = 0.588$$

$$P(T_{0010} \text{ fail} | 1010) = P(T_{0010} \text{ fail} | \text{LRU1 tested}) = 0.18$$

$$P(T_{0010} \text{ pass} | 1010) = P(T_{0010} \text{ pass} | \text{LRU1 tested}) = 0.82$$

These branch probabilities are placed on the BIT testing diagram in Figure 2.

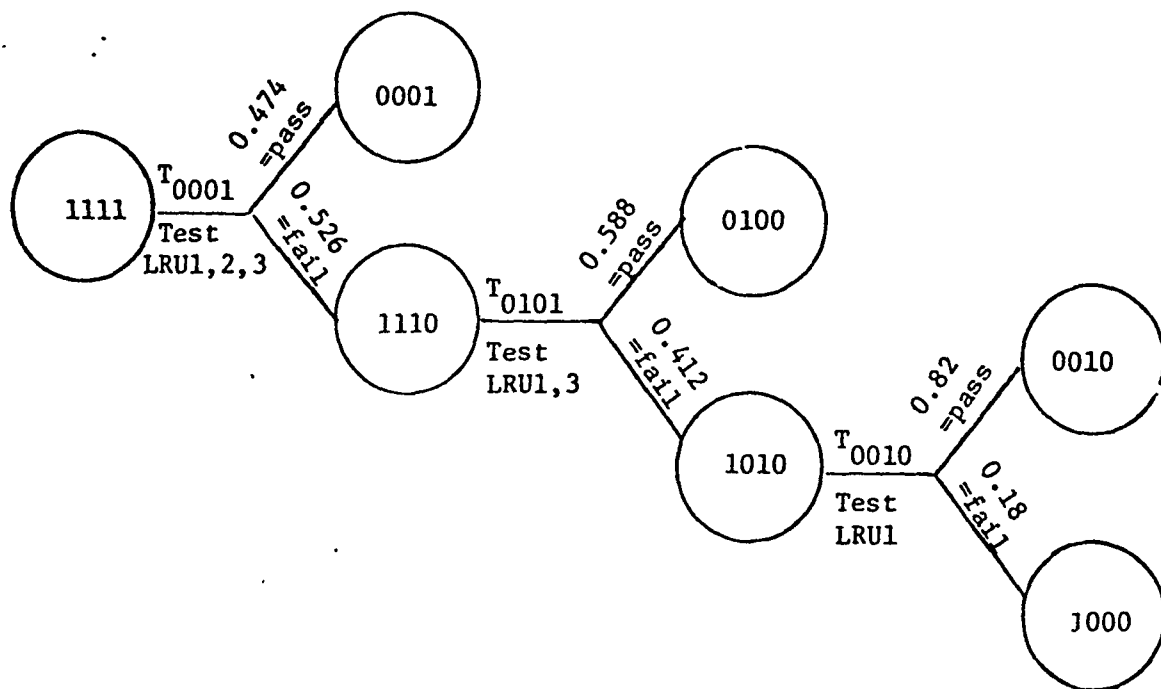


Figure 2. BIT Testing Diagram with Branch Probabilities

The state probabilities for the testing diagram are computed below.

$$P(1111) = 1$$

$$P(0001) = P(1111)P(T_{0001} \text{ pass} | 1111) = 1(0.474) = 0.474$$

$$P(1110) = P(1111)P(T_{0001} \text{ fail} | 1111) = 1(0.526) = 0.526$$

$$P(0100) = P(1110)P(T_{0101} \text{ pass} | 1110) = 0.526(0.588) = 0.309$$

$$P(1010) = P(1110)P(T_{0101} \text{ fail} | 1110) = 0.526(0.412) = 0.217$$

$$P(0010) = P(1010)P(T_{0010} \text{ pass} | 1010) = 0.217(0.82) = 0.178$$

$$P(1000) = P(1010)P(T_{0010} \text{ fail} | 1010) = 0.217(0.18) = 0.039$$

These state probabilities are placed on the testing diagram in Figure 3.

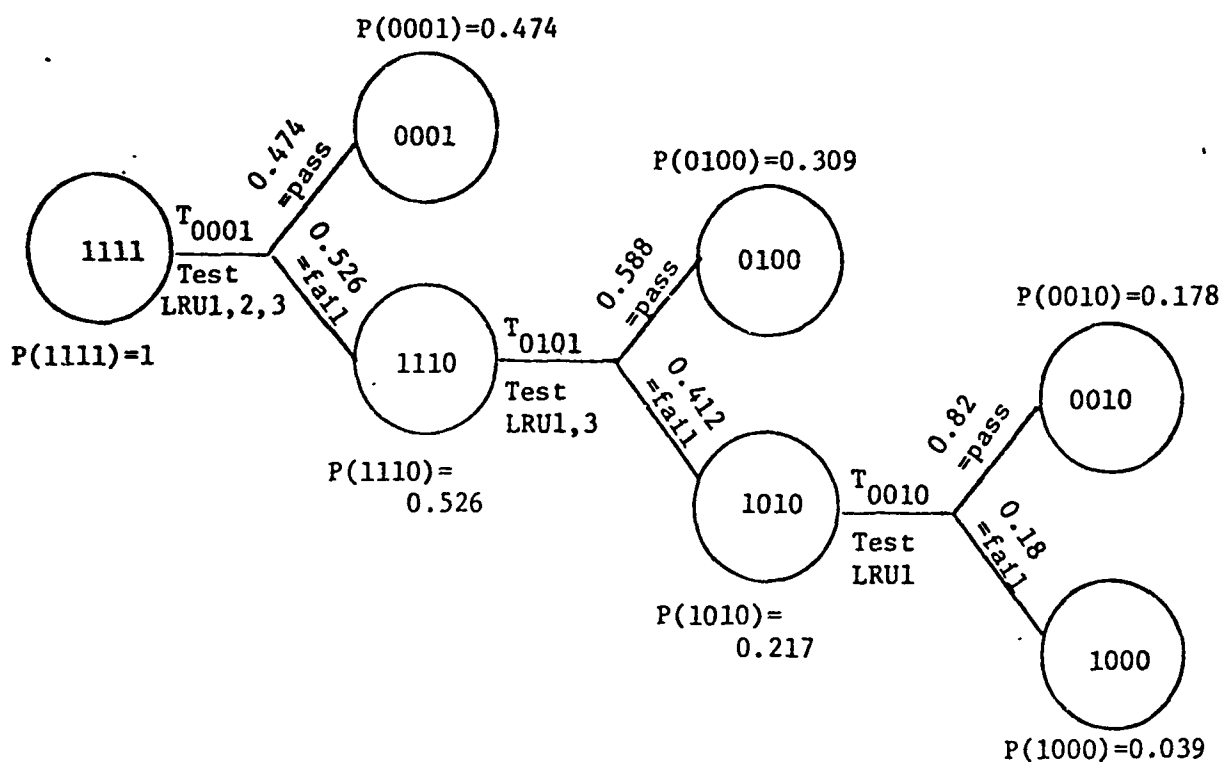


Figure 3. BIT Testing Diagram with Branch Probabilities and State Probabilities

The probabilities of the terminal states in the testing diagram shown in Figure 3 are:

$$P_1 = P(1000) = 0.039$$

$$P_2 = P(0100) = 0.309$$

$$P_3 = P(0010) = 0.178$$

$$P_4 = P(0001) = 0.474$$

Since the four terminal states in this testing diagram are mutually exclusive and exhaustive, $\sum_{j=1}^4 P_j = 1$.

LIFE CYCLE COST OF A BIT TESTING SEQUENCE

The objective of this research is to generate a sequence of BITs with a minimum life cycle cost. The life cycle cost of a BIT testing sequence is equal to the sum of the costs of the BITs plus the expected cost of secondary isolation for the mutually exclusive groups of LRUs that comprise the terminal states which can be called out by the BIT diagnostic. In equation form

$$LCC = \sum_{k=1}^M C_k + E[I_j(S_t)] \quad (8a)$$

$$= \sum_{k=1}^M C_k + \sum_{j=1}^K P_j I_j(S_t) \quad (8b)$$

where $M \leq 2^{N-1} - 1$

Equation (8) will be used to compute the life cycle cost of the BIT testing diagram given in Figure 3. That diagram reappears in Figure 4 with the state-dependent costs of the BITs, taken from Table 3, added to the diagram. Also added to the testing diagram in Figure 4 are the costs of secondary isolation for the terminal states, given in Table 2. State

probabilities appear in parentheses.

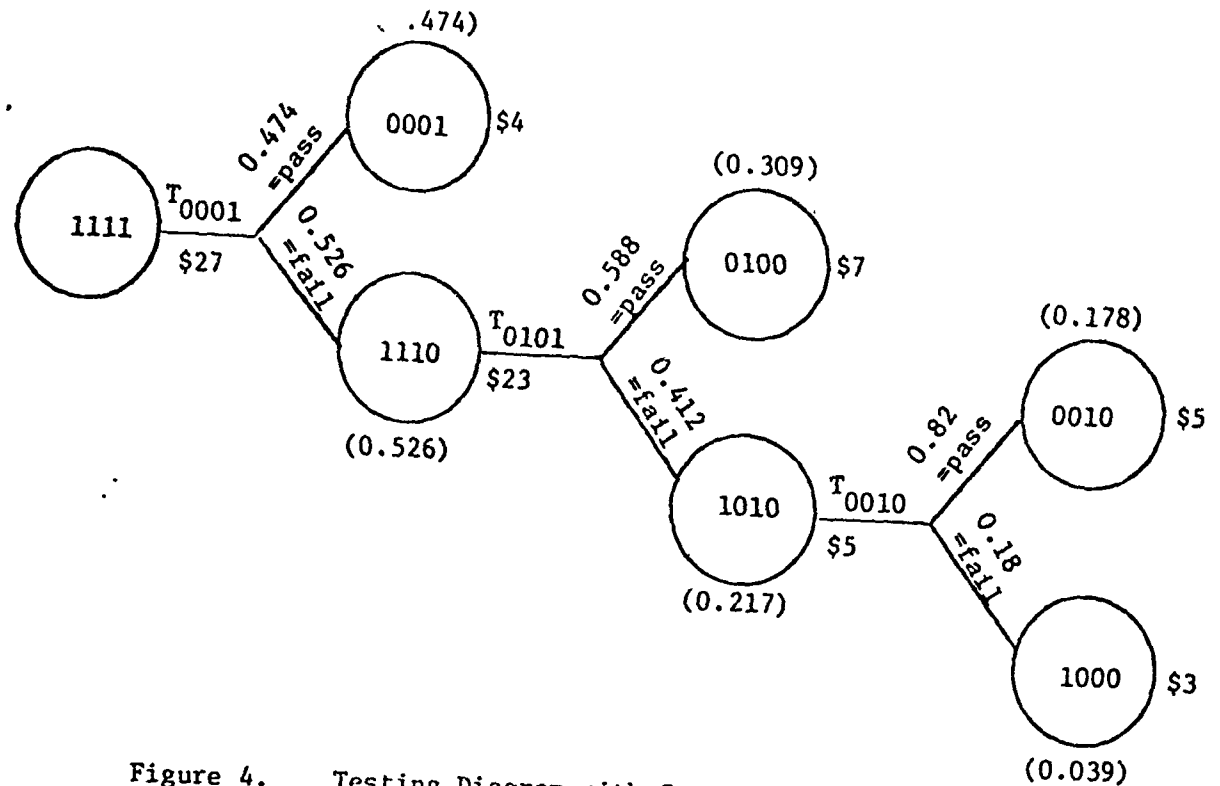


Figure 4. Testing Diagram with Costs of BITs and Costs of Secondary Isolation

Using equation (8b) to compute the life cycle cost of the BIT testing diagram in Figure 4,

$$\begin{aligned}
 LCC &= \sum_{k=1}^3 C_k + \sum_{j=1}^4 P_j I_j (S_t) \\
 &= [C_{0001|1111} + C_{0101|1110} + C_{0010|1010}] \\
 &\quad + [P_1 I_1 (1000) + P_2 I_2 (0100) + P_3 I_3 (0010) + P_4 I_4 (0001)] \\
 &= [\$27 + \$23 + \$5] + [0.039(\$3) + 0.309(\$7) + 0.178(\$5) + 0.474(\$4)] \\
 &= \$60.067
 \end{aligned} \tag{9}$$

RECURSIVE EQUATION FOR CALCULATING THE LIFE CYCLE COST OF A BIT TESTING DIAGRAM

A recursive equation can be formulated to calculate the life cycle cost of a BIT testing diagram. Let $f(S, T_k)$ be the life cycle cost of a sequence of BITs, given that the current state is S and BIT T_k is performed. Initially assume that test T_k produces two terminal states, symbolized by Y_t and Z_t . In this case, following the logic of equations (8b) and (7),

$$f(S, T_k) = C_{T_k}|S + \sum_{j=1}^2 P_j I_j(S_t) \quad (10)$$

where

$$P_j = \begin{cases} P(Y_t) = P(S)P(T_k \text{ pass}|S) & \text{if } T_k \text{ passes} \end{cases} \quad (11a)$$

$$P_j = \begin{cases} P(Z_t) = P(S)P(T_k \text{ fail}|S) & \text{if } T_k \text{ fails} \end{cases} \quad (11b)$$

and

$$I_j(S_t) = \begin{cases} I_j(Y_t) & \text{if } T_k \text{ passes} \end{cases} \quad (12a)$$

$$I_j(S_t) = \begin{cases} I_j(Z_t) & \text{if } T_k \text{ fails} \end{cases} \quad (12b)$$

Hence

$$f(S, T_k) = C_{T_k}|S + P(Y_t)I_j(Y_t) + P(Z_t)I_j(Z_t) \quad (13)$$

$$= C_{T_k}|S + P(S)[P(T_k \text{ pass}|S)I_j(Y_t) + P(T_k \text{ fail}|S)I_j(Z_t)] \quad (14)$$

Let

$$f(S) = \min_k \{f(S, T_k), I_j(S)\} \quad (15)$$

A recursive formulation for $f(S, T_k)$ expressed as a function of $f(Y)$ or $f(Z)$ is necessary whenever T_k does not produce two terminal states. Suppose that a single terminal state, Y_t , is produced only when T_k passes. However, when T_k fails assume that the resulting state, Z , is not a terminal state. Then

$$f(S, T_k) = C_{T_k|S} + P(S)P(T_k \text{ pass}|S)I_j(Y_t) + f(Z) \quad (16)$$

Similarly, if a terminal state, Z_t , is produced only when T_k fails,

$$f(S, T_k) = C_{T_k|S} + f(Y) + P(S)P(T_k \text{ fail}|S)I_j(Z_t) \quad (17)$$

Finally, if both Y and Z are non-terminal states,

$$f(S, T_k) = C_{T_k|S} + f(Y) + f(Z) \quad (18)$$

Equations (14), (16), (17) and (18) can be combined in a single equation (19) as

$$\begin{aligned} f(S, T_k) = & C_{T_k|S} + y f(Y) + (1-y)P(S)P(T_k \text{ passes}|S)I_j(Y_t) \\ & + z f(Z) + (1-z)P(S)P(T_k \text{ fails}|S)I_j(Z_t) \end{aligned} \quad (19)$$

where

$$y = \begin{cases} 1, & \text{if } Y \text{ is a non-terminal state} \\ 0, & \text{if } Y_t \text{ is a terminal state} \end{cases}$$

$$z = \begin{cases} 1, & \text{if } Z \text{ is a non-terminal state} \\ 0, & \text{if } Z_t \text{ is a terminal state} \end{cases}$$

The recursive equations (19) and (15) will be used to evaluate the life cycle cost of the BIT resting diagram in Figure 4.

$$\begin{aligned}
f(1010) &= C_{0010|1010} + P(1010)[P(T_{0010} \text{ pass}|1010)I_j(0010) \\
&\quad + P(T_{0010} \text{ fail}|1010)I_j(1000)] \\
&= \$5 + .217[.82(5) + .18(3)] = \$6.007
\end{aligned} \tag{20}$$

$$\begin{aligned}
f(1110) &= C_{0101|1110} + P(1110)P(T_{0101} \text{ pass}|1110)I_j(0100) + f(1010) \tag{21} \\
&= \$23 + .526(.588)7 + 6.007 = \$31.172
\end{aligned}$$

$$\begin{aligned}
f(1111) &= C_{0001|1111} + P(1111)P(T_{0001} \text{ pass}|1111)I_j(0001) + f(1110) \tag{22} \\
&= \$27 + 1(.474)4 + 31.172 = \$60.068
\end{aligned}$$

which agrees with the result obtained with equation (9).

RECURSIVE ALGORITHM FOR GENERATING A LEAST COST SEQUENCE OF BITS

Equations (19) and (15) can be used together to implement a recursive algorithm for generating a least cost sequence of BITS. The recursive algorithm consists of a backward solution procedure followed by a forward computational process. A testing sequence can be divided into stages, such that each stage represents the number of untested elements. At each state of every stage a set of possible decisions consists of all of the BITS which can be performed plus the alternative of performing secondary isolation of that state. The solution procedure begins by moving backward stage by stage. The backward solution procedure starts at stage one in which each state contains a single failed LRU. For each state S

of every stage evaluated during the backward solution procedure, no numerical value is known for $P(S)$ because the numerical value of $P(S)$ is dependent on a unique BIT testing diagram. However, the desired BIT testing sequence is determined only after completion of the entire recursive algorithm. Therefore, $f(S, T_k)$ must be expressed as a function of the unknown variable $P(S)$ during the backward recursive procedure.

The evaluation of $f(S)$ during the backward recursive algorithm is facilitated by the fact that $f(S, T_k)$ can be expressed as a linear function of the unknown variable $P(S)$. To understand why $f(S, T_k)$ is a linear function of $P(S)$, observe that every backward solution begins by considering potential terminal states at stage one. Consider equation (14) in which both Y_t and Z_t are terminal states.

$$f(S, T_k) = C_{T_k|S} + P(S) [P(T_k \text{ pass} | S) I_j(Y_t) + P(T_k \text{ fail} | S) I_j(Z_t)] \quad (14)$$

Except for $P(S)$, the numerical values of all of the terms on the right hand side of equation (14) are known. For example, in the four-element sample problem the numerical values of $C_{T_k|S}$, of $P(T_k \text{ pass} | S)$, and of $P(T_k \text{ fail} | S)$ can be found in Table 3. The numerical values of both $I_j(Y_t)$ and $I_j(Z_t)$ are specified in Table 2. After appropriate numerical values have been substituted in equation (14), $f(S, T_k)$ can be represented as a linear function of $P(S)$, as shown in equation (23).

$$f(S, T_k) = a_k + b_k P(S), \quad (23)$$

where a_k and b_k are constants.

For all states S , $0 \leq P(S) \leq 1$ because $P(S)$ is a probability. Therefore, it will generally be possible to find a unique value for $f(S)$,

expressed either numerically as the value of $I_t(S)$ which is known from Table 2, or expressed instead as a linear function of the unknown state probability $P(S)$. In the latter case

$$f(S) = \min_k \{f(S, T_k), I_j(S)\} = a_{k*} + b_{k*} P(S),$$

where the asterisk denotes the value of k which minimizes the function $f(S, T_k)$.

The backward recursive process ends at stage N in which there is a single initial state consisting of all 1's since no LRUs have been tested. The probability of the initial state is equal to 1 because every testing diagram must start with all of the LRUs untested. Since the probability of the initial state is known, all other state probabilities can be obtained by beginning at the starting state and successively applying equation (7) to calculate the probabilities of states at the lower numbered stages. By executing these forward computations of state probabilities, an optimal testing sequence can be constructed along with the partition into mutually exclusive terminal states which this testing sequence produces.

To calculate the state probabilities and identify an optimal testing sequence, which has its origin in the starting state, equations (7) are used in a forward computational process. In the initial state $S = 11 \dots 1$ and $P(11 \dots 1) = 1$. Following the performance of an optimal test, T_k^* , on the initial state, the two successor states, Y and Z , can be determined by using equation (7).

$$P(Y) = P(11 \dots 1)P(T_k^* \text{ pass} | 11 \dots 1) \quad (7a)$$

$$P(Z) = P(11 \dots 1)P(T_k^* \text{ fail} | 11 \dots 1) \quad (7b)$$

States Y and Z are nodes at the ends of the first upward sloping branch and the first downward sloping branch of an optimal testing diagram which originates from a starting state consisting of all 1's. Equation (7) is next applied to states Y and Z by performing the tests which are optimal in each of those states. This forward computational procedure is repeated at each new non-terminal successor state until an entire minimum cost testing diagram and the terminal states which it produces have been identified.

SOLUTION OF EXAMPLE BY RECURSIVE ALGORITHM

The complete recursive algorithm is demonstrated by applying it to the four-element sample problem specified in Tables 1 and 2. The backward recursive computations are summarized in Tables 4, 5 and 6. Figures 5, 6 and 7 are used to illustrate backward recursive calculations for representative entries in Tables 4, 5 and 6.

Table 4

Values of Terminal States at Stage One

n=1				
s_t	1000	0100	0010	0001
$I(s_t)$	\$ 3	7	5	4

Table 5

Computations for $f(S)$ using Equations (15) and (19)

	0001	0010	0011	0100	0101	0110	0111	I(S)	f(S)	T_k^*
				0, 0 0100, 1000 12+ 6.392 P(1100)	0, 0 0100, 1000 14+ 5.304 P(1100)	0, 0 0100, 1000 12+ 6.456 P(1100)	0, 0 0100, 1000 10+ 6.408 P(1100)	20	10+ 6.408 P(1100)	0111
		0, 0 0010, 1000 5+ 4.64 P(1010)	0, 0 0010, 1000 11+ 4.672 P(1010)			0, 0 0010, 1000 12+ 4.728 P(1010)	0, 0 0010, 1000 10+ 4.704 P(1010)	20	5+ 4.64 P(1100)	0010
	0, 0 0001, 1000 11+ 3.86 P(1001)		0, 0 0001, 1000 11+ 3.836 P(1001)		0, 0 0001, 1000 14+ 3.826 P(1001)		0, 0 0001, 1000 10+ 3.852 P(1001)	20	10+ 3.852 P(1001)	0111
		0, 0 0010, 0100 13+ 5.48 P(0110)	0, 0 0010, 0100 4+ 5.558 P(0110)	0, 0 0100, 0010 15+ 6.464 P(0110)	0, 0 0100, 0010 9+ 6.424 P(0110)			20	4+ 5.558 P(0110)	0011
	0, 0 0001, 0100 6+ 4.765 P(0101)		0, 0 0001, 0100 4+ 4.837 P(0101)	0, 0 0100, 0001 7+ 6.748 P(0101)		0, 0 0100, 0001 8+ 6.757 P(0101)		20	4+ 4.837 P(0101)	0011
	0, 0 0001, 0010 10+ 4.26 P(0011)	0, 0 0010, 0001 8+ 4.925 P(0011)			0, 0 0001, 0010 9+ 4.288 P(0011)	0, 0 0010, 0001 8+ 4.919 P(0011)		20	8+ 4.919 P(0011)	0110

Key to entries in a typical cell

 y, z
 Y, Z
 $f(S, r_k)$

Table 6

Computations for f(S) using Equations (15) and (19)

	0001	0010	0011	0100	0101	0110	0111	I(S)	f(S)	T _k *
	0, 1 0010, 1100 28+ 5.531 P(1110)	0, 1 0010, 1100 25+ 5.559 P(1110)	0, 1 0010, 1100 25+ 5.559 P(1110)	0, 1 0100, 1010 32+ 6.106 P(1110)	0, 1 0100, 1010 28+ 6.028 P(1110)	1, 0 0110, 1000 16+ 5.210 P(1110)	1, 0 0110, 1000 14+ 5.179 P(1110)	30	14+ 5.179 P(1110)	0111
0, 1 0001, 1100 27+ 4.864 P(1101)		0, 1 0001, 1100 25+ 4.956 P(1101)	0, 1 0001, 1100 25+ 4.956 P(1101)	0, 1 0100, 1001 29+ 6.298 P(1101)	1, 0 0101, 1000 18+ 4.517 P(1101)	0, 1 0100, 1001 30+ 6.352 P(1101)	1, 0 0101, 1000 14+ 4.565 P(1101)	30	14+ 4.565 P(1101)	0111
0, 1 0001, 1010 26+ 4.233 P(1011)	0, 1 0010, 1001 23+ 4.722 P(1011)	1, 0 0011, 1000 19+ 4.604 P(1011)			0, 1 0001, 1010 28+ 4.264 P(1011)	0, 1 0010, 1001 30+ 4.764 P(1011)	1, 0 0011, 1000 18+ 4.635 P(1011)	30	18+ 4.635 P(1011)	0111
0, 1 0001, 0110 20+ 4.700 P(0111)	0, 1 0010, 0101 25+ 4.952 P(0111)	1, 0 0011, 0100 12+ 5.500 P(0111)	0, 1 0100, 0011 30+ 6.315 P(0111)		1, 0 0101, 0010 13+ 4.884 P(0111)	1, 0 0110, 0001 12+ 5.432 P(0111)		30	12+ 5.432 P(0111)	0110
0, 1 0001, 1110 45.620	0, 1 0010, 1101 44.815	1, 1 0011, 1100 38.510	0, 1 0100, 1011 57.981		1, 1 0101, 1010 36.756	1, 1 0110, 1001 39.207	1, 0 0111, 1000 27.072	40	27.072	0111

Y, Z
Y, Z
f(S, T_k)

Key to entries in a typical cell

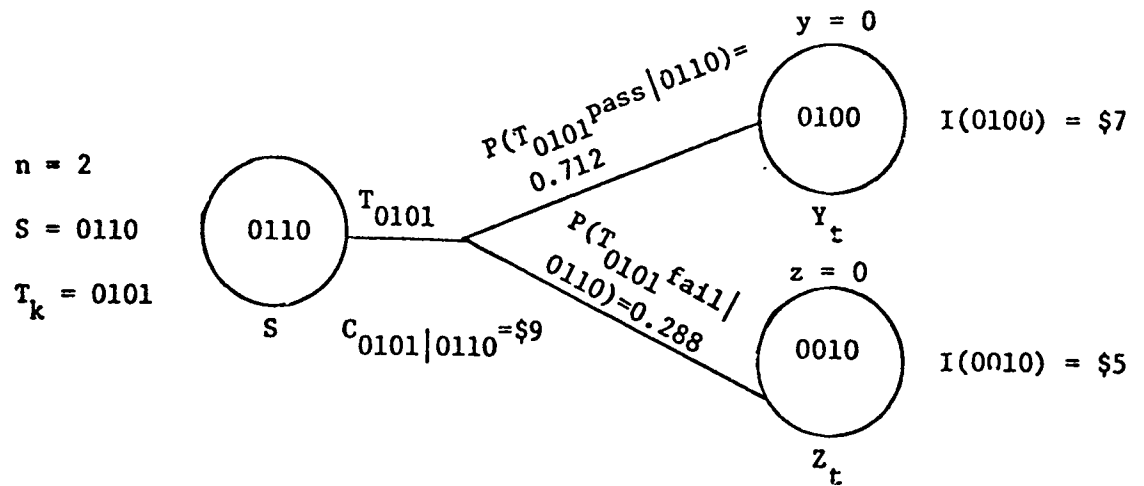


Figure 5. Testing Diagram for $f(0110, 0101)$

Calculations are indicated for $f(0110, 0101)$ and $f(0110)$ in Table 5. To evaluate $f(0110, 0101)$ as a linear function of the unknown state probability $P(0110)$, equation (14) is used because both $Y_t = 0100$ and $Z_t = 0010$ are terminal states.

$$\begin{aligned}
 f(0110, 0101) &= C_{0101|0110} + P(0110)[P(T_{0101} \text{ pass} | 0110)I(0100) \\
 &\quad + P(T_{0101} \text{ fail} | 0110)I(0010)] \\
 &= 9 + P(0110)[0.712(7) + 0.288(5)] \\
 &= 9 + 6.424P(0110)
 \end{aligned} \tag{14}$$

Equation (15) is used to find a unique value for $f(0110)$, in this case expressed as a linear function of $P(0110)$.

$$f(S) = \min_k \{f(S, T_k), I_j(S)\} \tag{15}$$

$$\begin{aligned}
 f(0110) &= \min\{f(0110, 0010), f(0110, 0011), f(0110, 0100), \\
 &\quad f(0110, 0101), I(0110)\}
 \end{aligned}$$

$$\begin{aligned}
&= \min\{13 + 5.48P(0110), 4 + 5.558P(0110), 15 + 6.464P(0110), \\
&\quad 9 + 6.424P(0110), 20\} \\
&= 4 + 5.558P(0110) @ T_k^* = T_{0011}
\end{aligned}$$

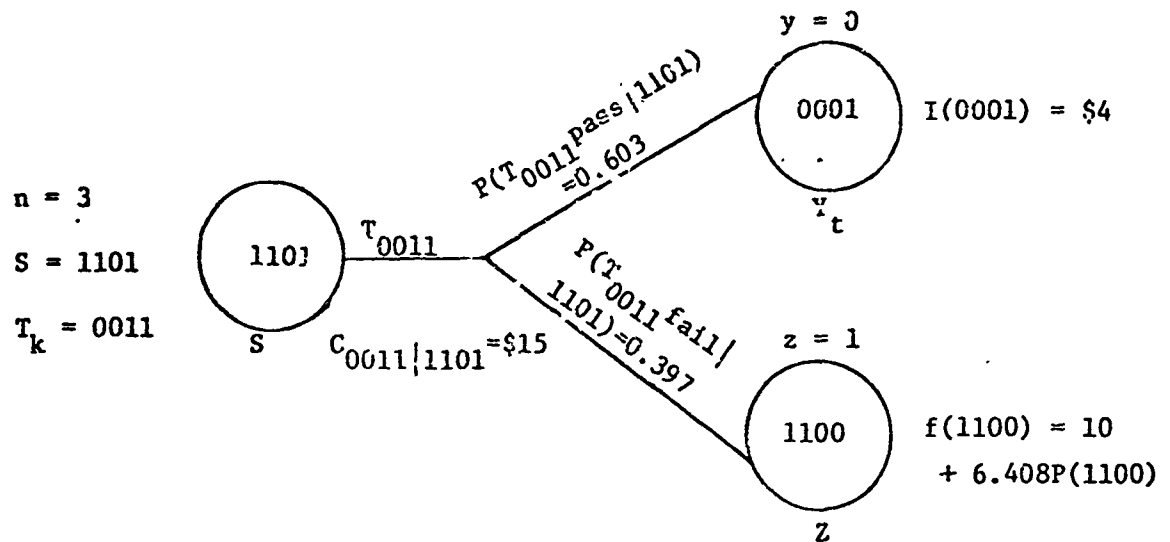


Figure 6. Testing Diagram for $f(1101, 0011)$

Calculations for $f(1101, 0011)$ and $f(1101)$ in Table 6 are illustrated below. Since $Y_t = 0001$ is a terminal state while $Z = 1100$ is not a terminal state, equation (16) is used to evaluate $f(1101, 0011)$.

$$\begin{aligned}
f(1101, 0011) &= C_{0011|1101} + P(1101)P(T_{0011} \text{ pass} | 1101)I(0001) \\
&\quad + f(1100)
\end{aligned} \tag{16}$$

From Table 5,

$$f(1100) = 10 + 6.408P(1100)$$

Using equation (7b) and Table 3,

$$P(1100) = P(1101)P(T_{0011} \text{ fail} | 1101) = 0.397P(1101) \tag{7b}$$

Substituting the value of $P(1100)$ gives

$$f(1100) = 10 + 6.408(0.397)P(1101)$$

and

$$\begin{aligned} f(1101, 0011) &= 15 + 0.603(4)P(1101) + [10 + 6.408(0.397)P(1101)] \\ &= 25 + 4.956P(1101) \end{aligned}$$

Equation (15) is used to find $f(1101)$ as a linear function of $P(1101)$.

$$\begin{aligned} f(1101) &= \min\{f(1101, 0001), f(1101, 0011), f(1101, 0100), f(1101, 0101), \\ &\quad f(1101, 0110), f(1101, 0111), I(1101)\} \\ &= \min\{27 + 4.864P(1101), 25 + 4.956P(1101), 29 + 6.298P(1101), \\ &\quad 18 + 4.517P(1101), 30 + 6.352P(1101), 14 + 4.565P(1101), 30\} \\ &= 14 + 4.565P(1101) \text{ at } T_k^* = T_{0111} \end{aligned}$$

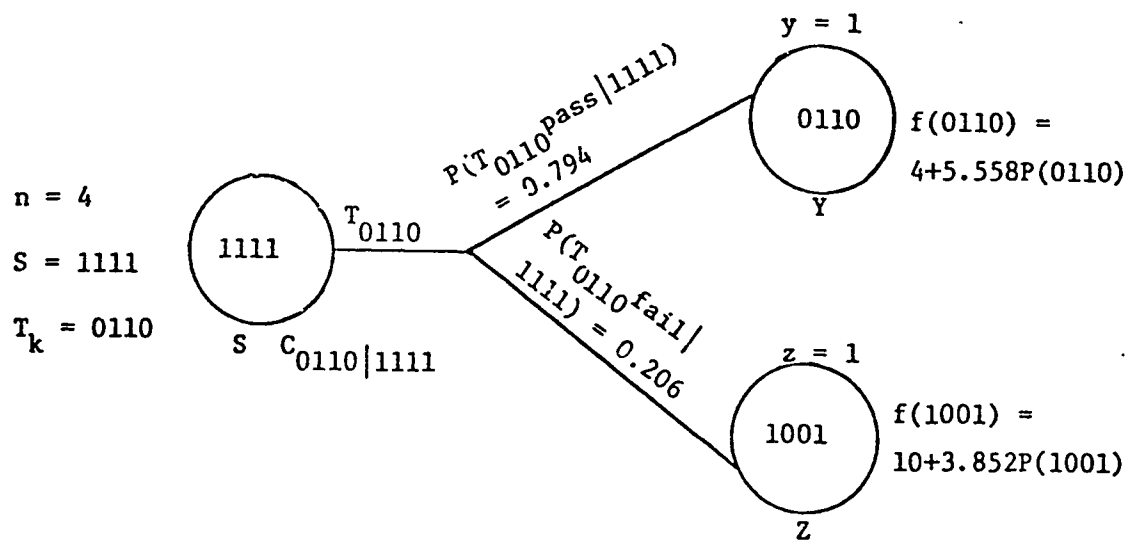


Figure 7. Testing Diagram for $f(1111, 0110)$

Calculations are given for $f(1111, 0110)$ and $f(1111)$ in Table 6. Since state $S = 1111$ is the initial state for the testing diagram, $P(1111) = 1$. Knowledge of the starting state probability enables numerical values to be calculated for $f(1111, 0110)$ and $f(1111)$. Furthermore, by moving forward from the starting state and repeatedly applying equations (7), the other state probabilities can be computed and thus the optimal testing diagram can be determined. Both $Y = 0110$ and $Z = 1001$ are non-terminal states. Therefore, equation (18) will be used to compute a numerical value for $f(1111, 0110)$.

$$f(1111, 0110) = C_{0110|1111} + f(0110) + f(1001) \quad (18)$$

From Table 5,

$$f(0110) = 4 + 5.558P(0110)$$

and

$$f(1001) = 10 + 3.852P(1001)$$

Using equations (7) and Table 3,

$$P(0110) = P(1111)P(T_{0110} \text{ pass} | 1111) = 1(0.794) \quad (7a)$$

$$P(1001) = P(1111)P(T_{0110} \text{ fail} | 1111) = 1(0.206) \quad (7b)$$

Substituting the values of $P(0110)$ and of $P(1001)$ in the equations for $f(0110)$ and $f(1001)$, respectively, gives

$$f(1111, 0110) = 20 + [4 + 5.558(0.794)] + [10 + 3.852(0.206)] = 39.207$$

Equation (15) is used to find a numerical value for $f(1111)$ which is the cost of the minimum cost testing diagram.

$$\begin{aligned}
 f(1111) &= \min\{f(1111, 0001), f(1111, 0010), f(1111, 0011), f(1111, 0100), \\
 &\quad f(1111, 0101), f(1111, 0110), f(1111, 0111), I(1111)\} \\
 &= \min\{45.620, 44.816, 38.510, 57.981, 36.756, 39.207, \\
 &\quad 27.072, 40\} \\
 &= 27.072 \text{ at } T_k^* = 0111
 \end{aligned}$$

Forward calculations of state probabilities are accomplished by successive applications of equation (7). States, state probabilities and costs in the optimum solution for the sample problem are summarized in Table 7.

Table 7

States, Tests, and Costs in the Optimum Solution
for the Sample Problem

Previous State	S Present State	P(S)	T_k^* test	f(S) cost	Y Next state if T_k^* passes	Z Next state if T_k^* fails
None	1111	1	T_{0111}	\$27.072	0111	1000
1111	0111	.852	T_{0110}	$12 + 5.432P(0111)$ =\$16.628	0110	0001
1111	1000	.148	$I_j(1000)$	$3P(1000)$ =\$0.444	none	none
0111	0110	.783	T_{0011}	$4 + 5.558P(0110)$ =\$8.352	0010	0100
0111	0001	.069	$I_j(0001)$	$4P(0001)$ =\$0.276	none	none
0110	0010	.565	$I_j(0010)$	$5P(0010)$ =\$2.825	none	none
0110	0100	.218	$I_j(0100)$	$7P(0100)$ =\$1.526	none	none

The optimum solution for the sample problem is the BIT testing sequence $\{T_{0111}, T_{0110}, T_{0011}\}$ with a life cycle cost of $f(1111) = \$27.072$. The minimum cost testing diagram is given in Figure 8.

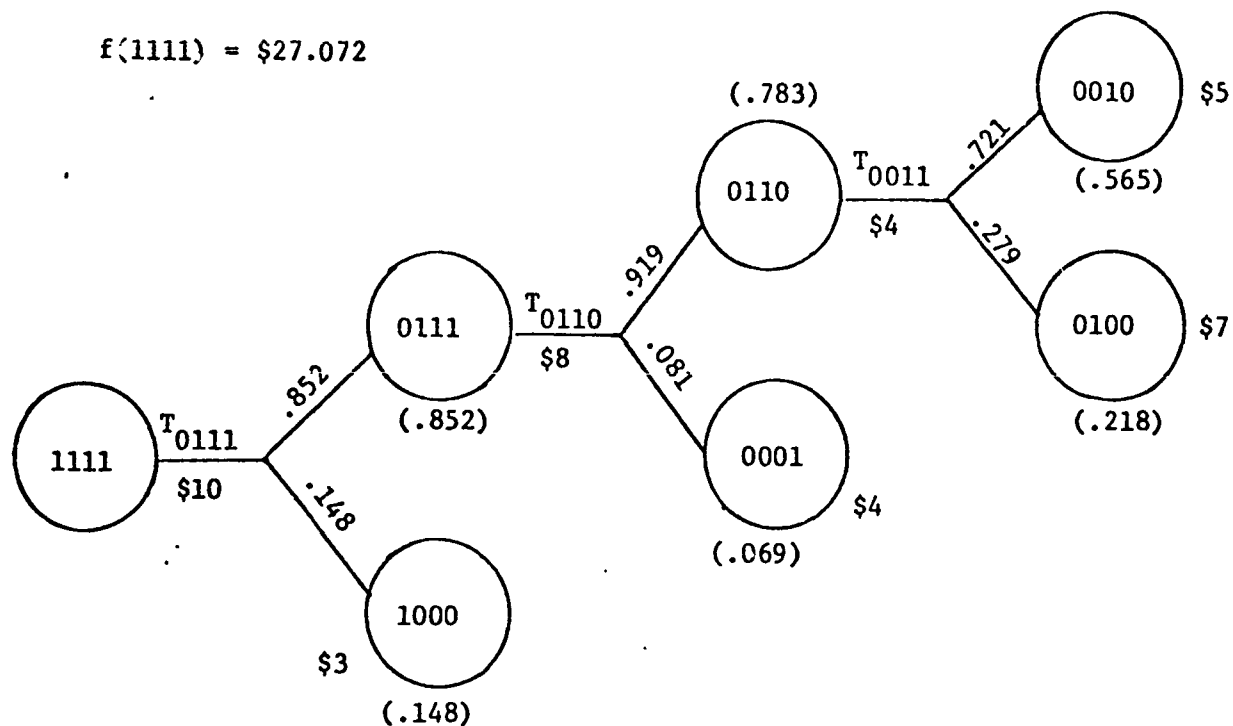


Figure 8. Optimum Testing Sequence for Sample Problem

When the life cycle cost of the optimum testing sequence is computed directly from the testing diagram in Figure 8 by using equation (8b),

$$C = (10+8+4) + 0.148(3) + 0.069(4) + 0.565(5) + 0.218(7) = \$27.071, \quad (24)$$

the result agrees with the life cycle cost derived by applying the recursive algorithm.

CONSTRAINTS

A minimum cost solution consisting of a sequence of BITS and the mutually exclusive terminal states called out by these BITS may be required to satisfy

constraints imposed by the Air Force. In this report the following three constraints are described, although other constraints could have been chosen.

1. A minimum proportion of equipment faults shall be detectable by the equipment's BIT capability.
2. A specified percentage of the repair times required to perform secondary isolation must not exceed a certain value.
3. The average number of maintenance manhours required to repair the equipment must not exceed a given value.

BIT DETECTION PROBABILITY

Consider an equipment consisting of N LRUs and a BIT diagnostic subsystem. The probability that the BIT subsystem will detect a faulty LRU is

$$(1-P_F) = \sum_{i=1}^N P(\text{LRU}_i=\text{bad})P(\text{BIT detects fault in LRU}_i | \text{LRU}_i=\text{bad})$$

$$= \sum_{i=1}^N p_i P(\text{BIT detects fault in LRU}_i | \text{LRU}_i=\text{bad}) \quad (25)$$

Consider the four-element sample problem defined in Tables 1 and 2. Suppose that a particular BIT testing sequence, $\{T_{0001}, T_{0101}, \text{ and } T_{0010}\}$, which is diagrammed in Figure 4, is implemented. To compute the BIT detection probability for this testing sequence, the probabilities that the BIT subsystem detects a fault in each LRU when that LRU is known to be bad must be computed.

$$P(\text{BIT detects fault in LRU4} \mid \text{LRU4} = \text{bad}) \quad (26)$$

$$= P(T_{0001} \text{ does not detect fault in LRU1, 2, or 3} \mid \text{LRU1, 2, 3} = \text{good})$$

$$= P(T_{0001} \text{ pass} \mid 1111, \text{LRU1, 2, 3} = \text{good}) = 1$$

$$P(\text{BIT detects fault in LRU2} \mid \text{LRU2} = \text{bad})$$

$$= P(T_{0001} \text{ detects fault in LRU1, 2, or 3} \mid \text{LRU2} = \text{bad})$$

$$P(T_{0101} \text{ does not detect fault in LRU1 or 3} \mid \text{LRU1, 3} = \text{good})$$

$$= P(T_{0001} \text{ fail} \mid 1111, \text{LRU2} = \text{bad}) P(T_{0101} \text{ pass} \mid 1110, \text{LRU1, 3} = \text{good}) \quad (27)$$

$$= 0.85(1) = 0.85$$

$$P(\text{BIT detects fault in LRU3} \mid \text{LRU3} = \text{bad})$$

$$= P(T_{0001} \text{ detects fault in LRU1, 2, or 3} \mid \text{LRU3} = \text{bad})$$

$$P(T_{0101} \text{ detects fault in LRU1 or 3} \mid \text{LRU3} = \text{bad})$$

$$P(T_{0010} \text{ does not detect fault in LRU1} \mid \text{LRU1} = \text{good})$$

$$\begin{aligned}
&= P(T_{0001} \text{ fail} | 1111, \text{LRU3} = \text{bad}) P(T_{0101} \text{ fail} | 1110, \text{LRU3} = \text{bad}) \\
&\quad P(T_{0010} \text{ pass} | 1010, \text{LRU 1} = \text{good}) \quad (28) \\
&= 0.65(0.72)1 = 0.468
\end{aligned}$$

$$P(\text{BIT detects fault in LRU1} | \text{LRU1} = \text{bad})$$

$$= P(T_{0001} \text{ detects fault in LRU1, 2, or 3} | \text{LRU1} = \text{bad})$$

$$P(T_{0101} \text{ detects fault in LRU1 or 3} | \text{LRU1} = \text{bad})$$

$$P(T_{0010} \text{ detects fault in LRU1} | \text{LRU1} = \text{bad})$$

$$= P(T_{0001} \text{ fail} | 1111, \text{LRU1} = \text{bad}) P(T_{0101} \text{ fail} | 1110, \text{LRU1} = \text{bad})$$

$$P(T_{0010} \text{ fail} | 1010, \text{LRU1} = \text{bad}) \quad (29)$$

$$= 0.7(0.87)0.9 = 0.548$$

The BIT detection probability for the testing sequence $\{T_{0001}, T_{0101}, T_{0010}\}$ is

$$\begin{aligned}
(1-P_F) &= p_1(0.548) + p_2(0.85) + p_3(0.468) + p_4(1) \\
&= 0.2(0.548) + 0.3(0.85) + 0.4(0.468) + 0.1(1) = 0.652 \quad (30)
\end{aligned}$$

Conversely, the probability that the BIT subsystem will not detect the group of LRUs containing the faulty unit is $P_F = 0.348$. In the latter event it will be necessary to manually troubleshoot the entire equipment to isolate the faulty LRU.

SPECIFIED PERCENTILE FOR THE DISTRIBUTION OF MAINTENANCE MANHOURS REQUIRED FOR SECONDARY ISOLATION

Assume the probability distribution of the maintenance manhours required for secondary isolation is known for all of the mutually exclusive groups of LRUs into which the equipment has been partitioned by its BIT diagnostic subsystem. An m^{th} percentile of the probability distribution of the random variable MG_j , representing the repair times required for the K mutually exclusive groups of LRUs, is a particular number of maintenance manhours, denoted by M_{max} , such that m percent of the time the equipment can be repaired in less than M_{max} hours. The following two inequalities must be satisfied:

$$P(MG_j \leq M_{\text{max}}) \geq m \cdot 10^{-2} \quad (31)$$

and

$$P(MG_j < M_{\text{max}}) \leq m \cdot 10^{-2} \quad (32)$$

To determine the m^{th} percentile of the distribution of MG_j , first rank the values of MG_j in ascending order, for $j = 1, 2, \dots, K$. Next, sum the probabilities of the ordered values of MG_j until M_{max} is identified.

Assume that the sample problem defined in Tables 1 and 2 has the BIT testing sequence $\{T_{0001}, T_{0101}, T_{0010}\}$ which is diagrammed in Figure 4. This BIT diagnostic subsystem isolates to the single failed LRU. The distribution of maintenance times required for secondary isolation can be obtained from Figure 4, which contains the probabilities that each of the four terminal states will be called out by the BIT diagnostic, and from

Table 2 which gives the time required to perform secondary isolation on each of these terminal states. The distribution of these maintenance times is displayed in Table 8.

Table 8

Distribution of Repair Times for Secondary Isolation

Group j	1	2	3	4
MG_j (hrs)	0.3	0.7	0.5	0.4
P_j	0.039	0.309	0.178	0.474

Suppose that 65% of the time the equipment must be repaired in less than 1.3 hours. To determine whether the indicated BIT testing sequence satisfies this specification, the 65th percentile of the distribution of repair times will be computed. In Table 9 the repair times are ranked in ascending order.

Table 9

Repair Times Ranked in Ascending Order

Group j	1	4	3	2
MG_j (hrs)	0.3	0.4	0.5	0.7
P_j	0.039	0.474	0.178	0.309

Summing the probabilities of the ranked repair times gives the following results:

$$\begin{aligned}
 P(MG_j \leq 1.2 \text{ hr}) &= P(MG_j = 0.3 \text{ hr}) + P(MG_j = 0.4 \text{ hr}) + P(MG_j = 0.5 \text{ hr}) \\
 &= 0.039 + 0.474 + 0.178 \\
 &= 0.691 \geq 0.65
 \end{aligned}
 \tag{33}$$

$$\begin{aligned}
 P(MG_j < 1.2 \text{ hr}) &= P(MG_j = 0.3 \text{ hr}) + P(MG_j = 0.4 \text{ hr}) \\
 &= 0.039 + 0.474 \\
 &= 0.513 \leq 0.65
 \end{aligned}
 \tag{34}$$

Therefore, the 65th percentile is $M_{\max} = 1.2 \text{ hr}$, and the given BIT testing sequence does meet the specification.

MEAN TIME TO REPAIR THE EQUIPMENT

If the automatic BIT diagnostic subsystem is capable of detecting the group of LRUs which contains the faulty unit, then secondary isolation must be performed to locate the failed LRU. The mean time to complete secondary isolation is

$$ME = E[MG_j] = \sum_{j=1}^K P_j MG_j$$

In the event that the automatic BIT subsystem does not identify the group of LRUs that contains the faulty unit, manual troubleshooting of the entire

equipment is required. The probability is P_F that manual troubleshooting will be required. The mean time to repair the equipment is:

$$\begin{aligned}
 M &= ME(1-P_F) + MF(P_F) \\
 &= (1-P_F) \sum_{j=1}^K P_j MG_j + (P_F)MF
 \end{aligned}
 \tag{35}$$

BIT FALSE ALARMS

If the BIT subsystem can generate false alarms by indicating an error in the equipment when no error is present, the previous analysis remains valid provided that the probabilities that the individual BITs will pass or fail are recomputed. To illustrate the calculation of the probabilities that a single BIT will pass or fail when false alarms are possible, consider the following example presented in Table 10 of a single BIT, T_{0011} , which can test any combination of LRUs 1 and 2.

Table 10
A BIT which can Generate False Alarms

Test	LRUk	P_k	$P(T_{0011} \text{ correctly detects fault} \text{LRUk bad})$	$P(T_{0011} \text{ incorrectly detects fault} \text{LRUk good})$
0011	1	0.2	0.82	0.06
	2	0.3	0.93	0.04

Assume that LRUs 1 and 2 are to be tested.

$$P(T_{0011} \text{ fail} | \text{LRU1, 2 tested})$$

$$\begin{aligned}
 &= P(\text{LRU1 bad}) P(T_{0011} \text{ correctly detects fault} | \text{LRU1 bad}) \\
 &+ P(\text{LRU2 bad}) P(T_{0011} \text{ correctly detects fault} | \text{LRU2 bad}) \\
 &+ P(\text{LRU1 good}) P(T_{0011} \text{ incorrectly detects fault} | \text{LRU1 good}) \\
 &+ P(\text{LRU2 good}) P(T_{0011} \text{ incorrectly detects fault} | \text{LRU2 good}) \\
 &- P(\text{LRU1 bad}) P(T_{0011} \text{ correctly detects fault} | \text{LRU1 bad}) \\
 &\quad P(\text{LRU2 bad}) P(T_{0011} \text{ correctly detects fault} | \text{LRU2 bad}) \\
 &- P(\text{LRU1 bad}) P(T_{0011} \text{ correctly detects fault} | \text{LRU1 bad}) \\
 &\quad P(\text{LRU2 good}) P(T_{0011} \text{ incorrectly detects fault} | \text{LRU2 good}) \\
 &- P(\text{LRU1 good}) P(T_{0011} \text{ incorrectly detects fault} | \text{LRU1 good}) \\
 &\quad P(\text{LRU2 bad}) P(T_{0011} \text{ correctly detects fault} | \text{LRU2 bad}) \\
 &- P(\text{LRU1 good}) P(T_{0011} \text{ incorrectly detects fault} | \text{LRU1 good}) \\
 &\quad P(\text{LRU2 good}) P(T_{0011} \text{ incorrectly detects fault} | \text{LRU2 good}) \quad (36)
 \end{aligned}$$

$$\begin{aligned}
 &= 0.2(0.82) + 0.3(0.93) + 0.8(0.06) + 0.7(0.04) - 0.2(0.82)0.3(0.93) \\
 &- 0.2(0.82)0.7(0.04) - 0.8(0.06)0.3(0.93) - 0.8(0.06)0.7(0.04) = 0.4!
 \end{aligned}$$

$$P(T_{0011} \text{ pass} | \text{LRU1, 2 tested})$$

$$\begin{aligned}
 &= P(\text{LRU1 good}) P(T_{0011} \text{ not detect fault} | \text{LRU1 good}) \\
 &\quad P(\text{LRU2 good}) P(T_{0011} \text{ not detect fault} | \text{LRU2 good}) \\
 &+ P(\text{LRU1 bad}) P(T_{0011} \text{ not detect fault} | \text{LRU1 bad}) \\
 &\quad P(\text{LRU2 good}) P(T_{0011} \text{ not detect fault} | \text{LRU2 good}) \\
 &+ P(\text{LRU1 good}) P(T_{0011} \text{ not detect fault} | \text{LRU1 good}) \\
 &\quad P(\text{LRU2 bad}) P(T_{0011} \text{ not detect fault} | \text{LRU2 bad}) \\
 &+ P(\text{LRU1 bad}) P(T_{0011} \text{ not detect fault} | \text{LRU1 bad}) \\
 &\quad P(\text{LRU 2 bad}) P(T_{0011} \text{ not detect fault} | \text{LRU2 bad}) \quad (37)
 \end{aligned}$$

$$\begin{aligned}
&= 0.8(0.94)0.7(0.96) + 0.2(0.18)0.7(0.96) + 0.8(0.94)0.3(0.07) \\
&+ 0.2(0.18)0.3(0.07) = 0.546 \\
&= 1 - P(T_{0011} \text{ fail} | \text{LRU}_{1,2} \text{ tested})
\end{aligned}$$

Once the probabilities of BIT false alarms have been included in the probabilities that each BIT will pass or fail, the recursive algorithm can be applied to derive a minimum cost BIT testing sequence.

SUMMARY

A recursive algorithm has been presented which will generate a minimum cost sequence of BITs. The BITs are imperfect tests in the sense that they may not detect all of the possible errors in the equipment, and they may generate false alarms by indicating an error when no error is present.

The proportion of faults detectable by each BIT is assumed to be known. The probabilities of false alarms produced by the BITs are also known. In addition, the conditional probabilities of failure of the LRUs, given that the equipment has malfunctioned, are known. Both the cost of a BIT and the probability that a BIT will pass or fail are functions of which LRUs remain to be tested. The cost of a BIT which tests a group of LRUs is the sum of the costs of implementing the BIT for each of the LRUs under test. The conditional probability that a BIT will pass or fail, given that certain LRUs are tested, is computed in terms of the conditional probabilities of failure of the LRUs, the probability that the BIT will detect an error in a faulty LRU, and the probability that the BIT will generate a false alarm. The life cycle cost of a BIT testing sequence is equal to the sum of the costs of the BITs plus the expected costs of secondary isolation for the mutually exclusive groups of LRUs that can be called out by the BIT diagnostic subsystem.

The recursive algorithm consists of a backward computational process followed by a forward computational process. During the backward process expected cost functions are computed recursively as linear functions of unknown state probabilities. (A state distinguishes untested LRUs from those which are known to be good.) During the forward process the state probabilities are evaluated and a sequence of BITs with a minimum life cycle cost is determined. The recursive algorithm is applied to a four-element sample problem to produce a numerical solution.

The following three constraints are described. 1. The BIT subsystem must be capable of detecting a minimum proportion of equipment faults. 2. A specified percentage of the repair times required to perform secondary isolation must not exceed a given value. 3. The mean time to repair the equipment must not exceed a specified value.

ACKNOWLEDGMENT

The author is grateful to the Air Force Office of Scientific Research, Air Force Systems Command, USAF, for sponsoring this research under Grant No. AFOSR-78-3496. He is grateful to Mr. Jerome Klion, Reliability Branch, Rome Air Development Center, Griffiss AFB, N.Y., for his interest in this work and for helpful discussions. The author also thanks Dr. Joseph Bram, Directorate of Mathematical and Information Sciences, Air Force Office of Scientific Research, for serving as Program Manager.

REFERENCES

- (1) Benowitz, N., D. F. Calhoun, G. E. Alderson, J. E. Bauer, and C. T. Joeckel, "An Advanced Fault Isolation System for Digital Logic," IEEE Trans. Comput., Vol. C-24, pages 489-497, May 1975.
- (2) Brule, J. D., R. A. Johnson, and E. J. Kletsy, "Diagnosis of Equipment Failures," IRE Transactions on Reliability and Quality Control, RQC-9, pages 23-34, April 1960.
- (3) Butterworth, R., "Some Reliability Fault-Testing Models," Ops. Res., Vol. 20, pages 335-343, March/April, 1972.
- (4) Chang, H. Y., "A Distinguishability Criterion for Selecting Efficient Diagnostic Tests," AFIPS Proc. of Spring Joint Computer Conference, 32, pages 529-534, 1968.
- (5) Chang, H. Y., E. G. Manning, and G. Metze, Fault Diagnosis in Digital Systems, Wiley - Interscience, 1970.
- (6) Cohn, M. and G. Ott, "Design of Adaptive Procedures for Fault Detection and Isolation," IEEE Transactions on Reliability, Vol. R-20, No. 1, pages 7-10, February 1971.
- (7) Firstman, S. I., and B. Gluss, "Optimum Search Routines for Automatic Fault Isolation," Ops. Res., Vol. 8, No. 4, July/August 1960.
- (8) Gluss, B., "An Optimum Policy for Detecting a Fault in a Complex System," Ops. Res., Vol. 7, pages 467-477, 1959.
- (9) Goldman, A. S. and T. B. Slattery, Maintainability, John Wiley & Sons, 1967.
- (10) Hiller, F. S. and G. J. Lieberman, Introduction to Operations Research, Holden-Day, 1974.
- (11) Johnson, R. A., E. J. Kletsy, and J. D. Brule, "Diagnosis of Equipment Failures," SURI, Rept. No. EE577-594T1, April 1959. AD-213876.
- (12) Maintainability Engineering Design Notebook, Revision II, and Cost of Maintainability. Martin Marietta Aerospace Corp. RADC-TR-74-308, Vol. I-III, January 1975.
- (13) Military Standard. On-Aircraft, Fault Diagnosis, Sub-Systems, Analysis/Synthesis Of. MIL-STD-1591, January 1977.

- (14) Sheskin, T. J., "Design of System Diagnostic and Fault Isolation Procedures," 1977 USAF/ASEE Summer Faculty Research Program, Vol. 2, AFOSR-TR-78-0349, AD-A051514, pp. 30-1 to 30-28.
- (15) Sheskin, T. J., "Fault Isolation for Modular Electronic Equipment," Proceedings Annual Reliability and Maintainability Symposium, 1979 January, Washington DC, published by IEEE.
- (16) Sheskin, T. J., "Partitioning of Modular Equipment for Fault Isolation," Microelectronics and Reliability, Pergamon Press, Vol. 17, No. 6, 1978, pp. 597-600.
- (17) Sheskin, T. J., "Probabilistic Dynamic Programming for Fault Isolation," Proceedings of the Workshop on the Information Linkage Between Applied Mathematics and Industry, Naval Postgraduate School, Monterey, California, February 23-25, 1978, Academic Press (in press), 7 pages.
- (18) Sheskin, T. J., "Sequencing of Diagnostic Tests for Fault Isolation by Dynamic Programming," IEEE Transactions on Reliability, Vol. R-27, No. 4, December 1978.
- (19) Sheskin, T. J., "Specification of Built-in-Tests for Modular Equipment," submitted on May 2, 1978, to Mr. J. Klion, Rome Air Development Center, Griffiss Air Force Base, N.Y., and to Dr. J. Bram, Air Force Office of Scientific Research, Washington, D.C., submitted under Grant AFOSR 78-3496 30 pages. Also submitted on July 25, 1978, to Mr. I. Krulac, Rome Air Development Center, Griffiss AFB, N.Y.